**DATA WAREHOUSING WITH IBM CLOUD COMPUTING DB2**

**PROJECT OBJECTIVES:**

Start building the data warehouse using IBM cloud Db2 warehouse. Define the schema and structure of the data warehouse tables. Identify data sources ( e.g., CSV files, databases)and design a strategy to integrate them into the data warehouse

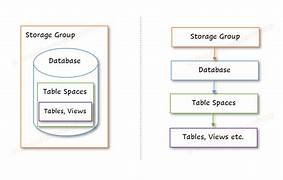
**INTRODUCTION:**

In the era of big data, businesses are faced with the challenge of managing and extracting valuable insights from vast amounts of information. To address this challenge, building a data warehouse is a strategic move. Data warehousing allows organizations to centralize, store, and efficiently analyze data from various sources, enabling data-driven decision-making. In this project, we will explore the process of building a data warehouse using IBM Cloud Db2 Warehouse, defining the schema and structure of the data warehouse tables, identifying data sources, and designing a strategy to integrate them into the data warehouse.

**STEPS IN DB2 TABLE:**

**1. Set Up IBM Cloud Db2 Warehouse:**

Before you start building your data warehouse, you need to set up an IBM Cloud Db2 Warehouse instance. You can do this through the IBM Cloud console.

****

**2. Define Schema and Structure:**

In Db2 Warehouse, you can define the schema and structure of your data warehouse using SQL. Here's an example of how to create a simple schema and table:

**PROGRAM:**

-- Create a new schema

CREATE SCHEMA my\_data\_warehouse;

-- Define a table in the schema

CREATE TABLE my\_data\_warehouse.my\_table (

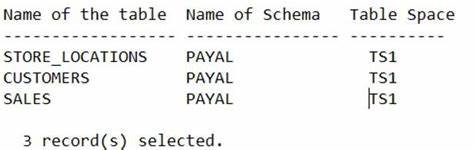
column1 INT,

column2 VARCHAR(50),

column3 DATE

);

**OUTPUT:**



**3. Identify Data Sources:**

Identify your data sources, which can include CSV files, databases, and other data repositories. For this example, let's assume you have CSV files and a remote database as your data sources.

**4. Data Integration Strategy:**

**a. Load Data from CSV Files:**

You can use Db2's IMPORT statement to load data from CSV files into your data warehouse table. Here's an example of how to do this:

**PROGRAM:**

-- Load data from a CSV file into the table

IMPORT FROM 'your\_file.csv'

OF DEL MODIFIED BY COLDEL, TIMESTAMPFORMAT="YYYY-MM-DD HH:MI:SS"

INSERT INTO my\_data\_warehouse.my\_table;

**b. Connect to External Databases**:

To integrate data from external databases, you can use Db2's FEDERATED feature. This allows you to connect to remote databases and query them as if they were local tables. First, you'll need to create a wrapper and a server definition for the external database:

**PROGRAM-1:**

-- Create a nickname to reference a remote table

CREATE WRAPPER DRDA;

CREATE SERVER remote\_db TYPE DB2/UDB VERSION 11.1 WRAPPER DRDA OPTIONS (

ADD DBNAME 'your\_remote\_db\_name',

ADD USER 'your\_db\_user',

ADD PASSWORD 'your\_db\_password',

ADD HOST 'remote\_db\_host',

ADD PORT 'remote\_db\_port'

);

Now, you can create a nickname to reference a table in the remote database:

**PROGRAM-2:**

-- Create a nickname to reference a remote table

CREATE NICKNAME my\_external\_table

FOR remote\_db.your\_schema.your\_remote\_table;

Once you've created the nickname, you can query the remote table as if it were a local table in your data warehouse.

**5. Schedule Data Integration:**

To keep your data warehouse up to date, you can schedule regular data integration jobs to import data from CSV files or refresh data from external databases. You can use scheduling tools or CRON jobs to automate this process.

**6. Data Warehousing Best Practices:**

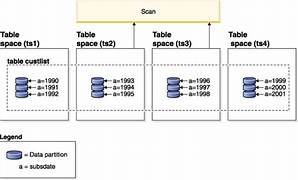
Monitor and optimize your data warehouse performance regularly.

Create indexes on columns frequently used in queries.

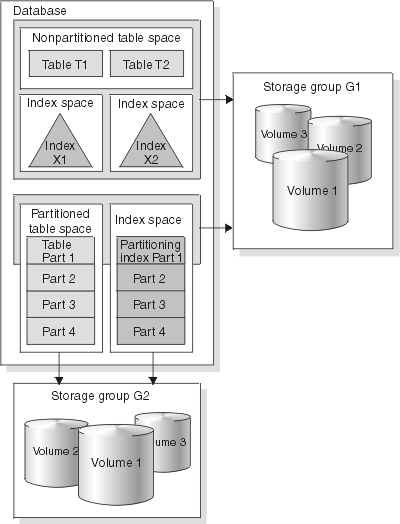
Consider partitioning tables to improve query performance on large datasets.

Implement data security and access controls.

This is a high-level overview of building a data warehouse using IBM Cloud Db2 Warehouse. Your specific implementation may vary depending on your data sources, data volume, and business requirements. Be sure to refer to the Db2 documentation and consult with IBM Cloud support for any specific details and optimizations.



Implementing a data warehouse, including ETL processes, schema creation, and data integration, typically involves a combination of tools and programming languages. Below, I'll provide a simplified example using Python and SQL to demonstrate a basic ETL process. Keep in mind that real-world scenarios can be much more complex.



For this example, we will consider two data sources: CSV files for sales and customer data. We'll use Python for data extraction, transformation, and loading (ETL) and SQL for creating and managing tables in the IBM Db2 Warehouse. You should have the necessary Python libraries installed, like pandas for data manipulation and the ibm\_db library for connecting to IBM Db2 Warehouse.

Please note that the exact implementation details, such as connection strings and file paths, will depend on your specific setup and environment.

**CODING:**

# Import necessary libraries

import pandas as pd

import ibm\_db

# Database connection parameters

db\_credentials = {

'dsn': 'your\_dsn',

'uid': 'your\_username',

'pwd': 'your\_password'

}

# Data source file paths

sales\_data\_csv = 'sales\_data.csv'

customer\_data\_csv = 'customer\_data.csv'

# Function to perform ETL

def etl\_process():

# Extract data from CSV files

sales\_data = pd.read\_csv(sales\_data\_csv)

customer\_data = pd.read\_csv(customer\_data\_csv)

# Transformation: Convert date strings to date objects, handle missing values, etc.

# Example transformations:

# sales\_data['Order Date'] = pd.to\_datetime(sales\_data['Order Date'])

# sales\_data.fillna(0, inplace=True)

# Establish a database connection

conn = ibm\_db.connect(f"{db\_credentials['dsn']};UID={db\_credentials['uid']};PWD={db\_credentials['pwd']}", "", "")

# Create tables if they don't exist (You need to define the table schemas)

create\_sales\_table = """

CREATE TABLE Sales (

OrderID INT PRIMARY KEY,

CustomerID INT,

ProductID INT,

OrderDate DATE,

Quantity INT,

SalesAmount DECIMAL(10, 2)

)

"""

create\_customer\_table = """

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY,

CustomerName VARCHAR(255),

Email VARCHAR(255),

Address VARCHAR(255)

)

"""

stmt = ibm\_db.exec\_immediate(conn, create\_sales\_table)

stmt = ibm\_db.exec\_immediate(conn, create\_customer\_table)

# Load data into the database (you can use SQL INSERT statements or a library like SQLAlchemy)

sales\_data.to\_sql('Sales', conn, if\_exists='append', index=False)

customer\_data.to\_sql('Customer', conn, if\_exists='append', index=False)

# Close the database connection

ibm\_db.close(conn)

# Run the ETL process

etl\_process()

This is a simplified example to get you started. In a real-world scenario, you would add error handling, logging, and more complex transformations and data sources. Additionally, you may need to schedule this ETL process to run periodically.

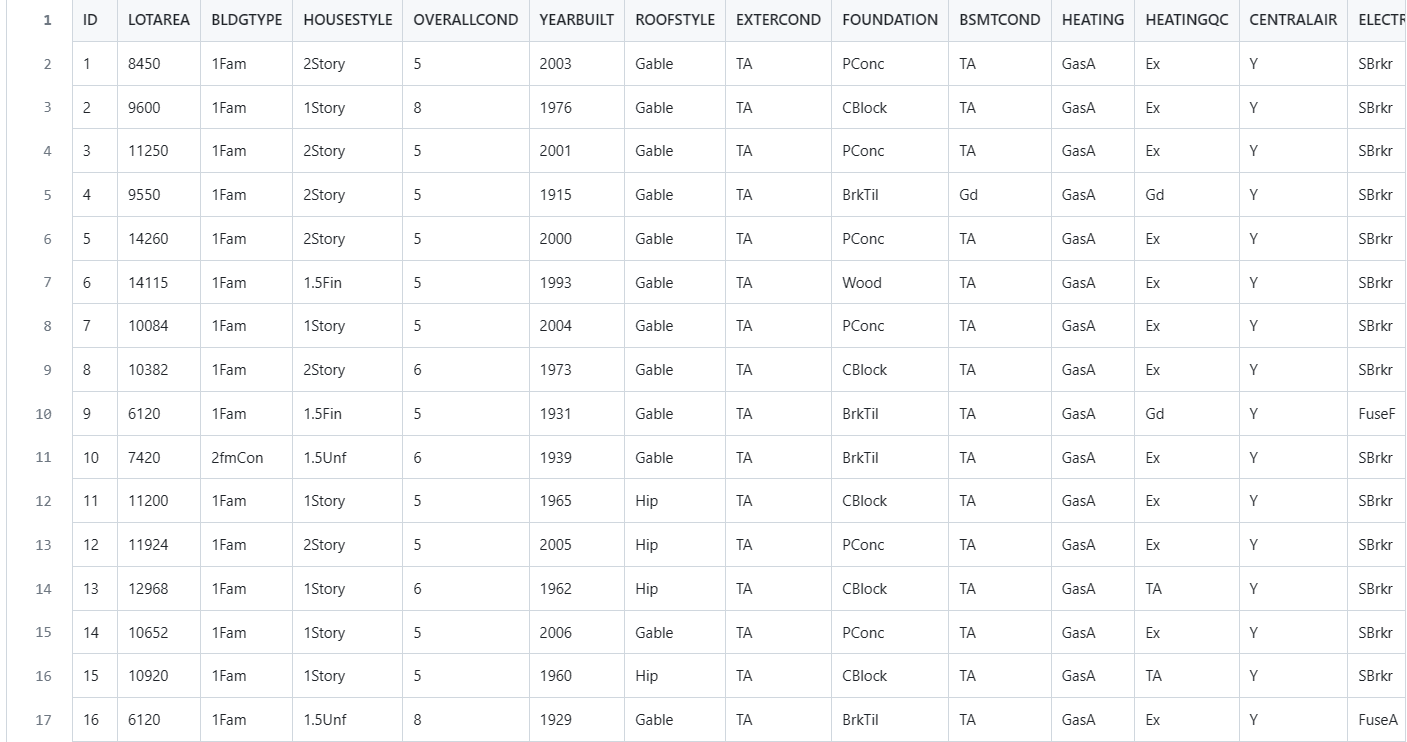
**OUTPUT:**

A diagram of a data flow

Description automatically generated

A diagram of a data flow

Description automatically generated

**EXAMPLE DATASET: **

**CONCLUSION:**

Building a data warehouse using IBM Cloud Db2 Warehouse is a strategic move for organizations . By defining a solid schema, identifying data sources, and designing a robust data integration strategy, organizations can unlock valuable insights from their data and gain a competitive edge in today's data-driven world. This project will guide you through these essential steps, helping you create a powerful and scalable data warehousing solution.

**TEAM MEMBERS:**

BHUVANESHWARI. P

PAVITHRA. A

RAJESHWARI. R

SHAMSHUN NAFIYA. N